Int J Syst Assur Eng Manag (December 2016) 7(Suppl. 1):S282–S293
DOI 10.1007/s13198-015-0403-1

ORIGINAL ARTICLE

Automated ontology generation from a plain text using statistical and NLP techniques

Naresh Kumar[1,2] · Minakshi Kumar[1,2] · Manjeet Singh[1,2]

Received: 26 October 2015 / Published online: 10 December 2015

Abstract Major portion of web pages contains the natural language text and understanding of natural language text from the web pages is a major challenge for machines. Due to this lacking search engines are not able to provide relevant information to the users. This problem is tackled by natural language processing techniques and the development of ontologies from natural language text. With the help of such ontologies search of information can increases manifold. Much research in the field of text processing and automatic ontology building from text has been done to address these challenges. The proposed method in this paper is another effort to build automatic ontology from domain specific text. In this method we first extract concepts from a given domain specific text. We have used a Stanford parser to parse the text and the dictionary of basic concepts is created manually containing all the domain specific concepts and their relationships by recognizing laxico-syntactic patterns in the text corpus. Once concepts and relations among concepts as well as properties of concepts are identified, the extracted information can be represented in the form of graph and OWL.

✉ Naresh Kumar
naresh.dhull@gmail.com

Minakshi Kumar
minakshiu007@gmail.com

Manjeet Singh
mstomer2000@yahoo.com

[1] AIIT, Amity University, Noida, India

[2] YMCA University of Science and Technology, Faridabad, India

Keywords Ontology · Natural language · OWL · Information retrieval · Statistical techniques

1 Introduction

Now-a-days, accessing information on the web is increasing at a tremendous rate. Most of the data available on the web is in the form of text written in natural language. Hence to retrieve the relevant information from web as well as text documents, there is a need to understand the text contained in them. As per a critical look at text processing, many challenges has been found which makes the task of text understanding difficult. In spite of these challenges, text understanding is gaining more and more importance during the last decades. Many techniques for text understanding like natural language processing (Fromkin et al 2000; ADen 2004), Ontologies to extract and understand the concepts from a text in a given document have been employed in different domains of speech synthesis and speech recognition etc. (Allen 1993; Jurafsky and Martin 2000). But natural language is highly ambiguous contains different type of ambiguities like Lexical ambiguity, Syntactic ambiguity, Semantic ambiguity as described by Dorr et al. (1998). Understanding of information structure and sharing its common understanding among people or software agents is one of the more common goals in developing ontologies (Gruber 1995). NLP and ontology together have shown a great improvement in extracting the knowledge from web pages. In literature survey various approaches for concept extraction and external knowledge based system for ontology building have been studied. Statistical and probability-driven based techniques are used for concept extraction these approaches aim to find the relevant domain specific concepts from the text corpus. External knowledge based ontology methods uses either WordNet (Moldovan and Girju 2000; Cho et al. 2006; http://wordnet.princeton.edu), Intranet (Kietz et al. 2000; http://enwikipedia.orglwikilIntranet) or WWW (Agirre et al. 2000; http://en.wikipedia.orgl/wiki/World_Wide_Web) as knowledge base. In this paper a different approach i.e. an automatic ontology generation for domain specific text is proposed. The implementation and result analysis of the proposed method have carried out and it is observed that implemented system is able to extract and represent the semantic knowledge in the form of concepts of ontology. The rest of this paper is organized as follows: In Sect. 2, related literature is discussed. In Sect. 3, we present our proposed scheme with detail architecture of the proposed system. In Sect. 4, corpus analysis for rule specification is discussed, In Sect. 5, the implementation of the system is presented and finally in Sect. 6, conclusion and future work is discussed.

2 Literature survey

In past few years extracting concepts from text of web pages and their representation in the form of ontology have grown to a whopping stage. Many researchers have applied different approaches to build an automatic ontology from text of web pages. Supervised learning is the most commonly used technique based on statistical and probability-driven analysis. This technique is also dependent on the use of a dictionary. Another approaches such as unsupervised learning uses intelligent algorithms such as cognitive and linguistically driven analysis to extract relevant concepts without the use of a large dictionary. As no single technique is completely perfect so a mixed model approach are used by many who work in the domain based on semantic analysis and they generate very good results. By combining linguistics with statistical analysis, they are able to eliminate the majority of the limitations of both supervised and unsupervised techniques. Ontology generation and classification have also gained popularity and grouped into four main categories (from the point of view of automation) of conversion or translation, mining based, external knowledge based and frameworks (Bedini and Nguyen 2007). Kong et al. (2006) have developed a technique based on WordNet in this method, to build a domain ontology subset of “concepts” are extracted by using WordNet as a general ontology. Moldovan and Girju (2000) also explained a method for generating ontologies based on WordNet. The approach is almost the same as given by Kong et al. (2006) but with difference. Here, designer defines some “seed”, concepts of the domain, and if a word is not found in WordNet then an additional module will look for it over the Internet. Agirre et al. (2000) have developed a strategy to enrich existing ontologies with the help of WWW to obtain the new information. Problem of proximity between two ontologies was used as choice between alignment and merging by authors in Cho et al. (2006). A generic approach for the domain based ontology creation which uses a domain based on source concepts with many entries was given by Kietz et al. (2000).

3 Proposed scheme

Basic idea behind our proposed scheme is that if we are able to comprehend the syntax of natural language text, the semantics of the text can be extracted from that syntactic structure. As it is our hypothesis that the syntactic and the semantic structure are interconnected to each other. In fact, as proposed by Chomsky (2002), the semantics in a sentence is the deep representation of the idea to be communicated whereas syntax is its surface representation.

Therefore, the proposed approach is based on the idea that if we can extract some patterns at syntactic level along with how they are realizing the corresponding semantics and if we can represent these syntactic semantic relationships or patterns in the form of rules, then these patterns can be used to find the semantic relationships between various concepts in given text provided we are capable of extracting syntactic structures using some natural language processing tool. So in our proposed scheme we first extracted such linguistic patterns on the basis of manual analysis of the text from a given domain. The patterns extracted are then represented using dictionaries and rules which jointly makes a sort of ontology. This base ontology of syntactic and semantic patterns are used find the semantic relationships between the concepts in a given test input. The final, semantic structured obtained this way are represented using graph structure and OWL as depicted in Fig. 1. It may be noted that in order to extract the various constituents and phrase types in a sentence of input text, we have to perform, some statistical analysis. In our proposed scheme we are using Stanford Parser (which perform statistical analysis) in order to extract the syntactic structure. Therefore, we are not performing any statistical analysis of text explicitly.

Even parsing of carpus text is also done to obtain the lexico-syntactic structure and these structures are, indeed, used to construct the syntactic-semantic rules manually. More specifically, based on this manual analysis different rules are made to identify the patterns for concept extraction, relation extraction, extraction of properties related to the concepts, finding subjects and objects of the verbs. The extracted information after analysis is stored in the appropriate structures as explained in the coming sections.

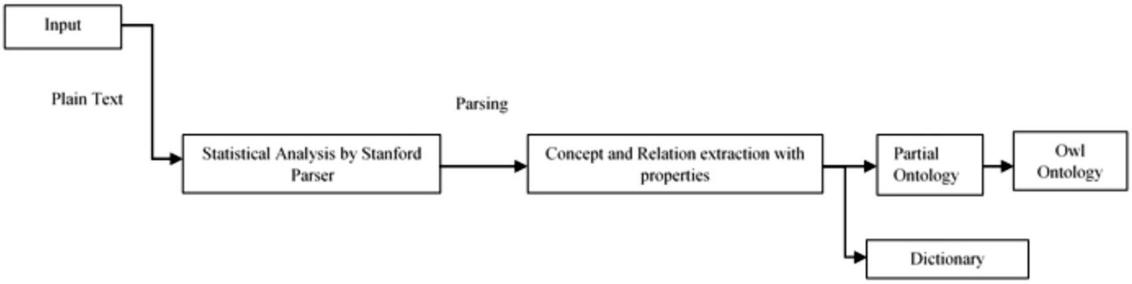

Input
Plain Text
Parsing
Statistical Analysis by Stanford Parser
Concept and Relation extraction with properties
Partial Ontology
Owl Ontology
Dictionary

Fig. 1 Proposed scheme for automatic ontology generation

3.1 Architecture of proposed method

In order to realize the proposed scheme, the following architecture is being proposed as depicted in Fig. 2. For this architecture, it is assumed that we have already constructed the syntactic-semantic structures based on manual analysis. The architecture contains following functional components:

- Concept extractor
- Hierarchical relationship extractor
- Properties extractor
- Action extractor

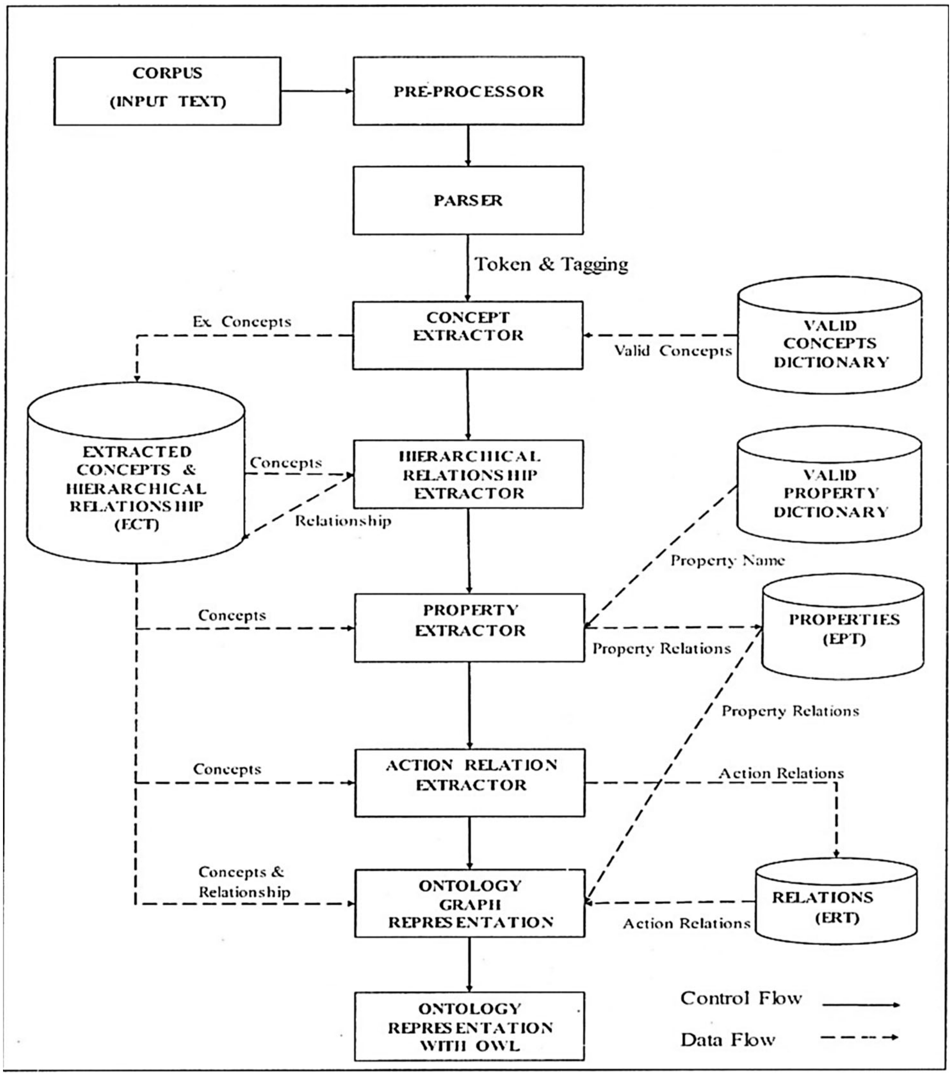


Fig. 2 Architecture of proposed system

- Graph representation
- Owl representation

3.1.1 Pre-processor

Pre-processing phase is the initial phase and makes text enable to be processed. This phase includes removal of unnecessary words called stop words.

3.1.2 Parser

After pre-processing phase text is given to parser which completes the tokenization of text in sentence, parsing and morphological analysis to bring it into singular form.

3.1.3 Concept extractor

Concept extractor find valid concepts including single word concept, multi-word concepts and nested concepts. Concept extractor extracts the concepts with the help of dictionary of recognizable concepts and rules. Concept extractor has following parts: Noun phrase extractor, Concept formation module, Concept Validator. First of all the noun phrases are extracted from the text. Then the noun phrase is analysed to form concepts. Each formed concept is validated with VCD (Dictionary).

3.1.4 Hierarchical relationship extractor

After the extraction of concepts hierarchical relationship extractor extract the hierarchical relationship between concepts which is being captured by two different ways viz. Inherent relationship in nested concepts and Hierarchical relationship defined by linking verb ‘is’.

3.1.5 Property extraction phase

Property extractor will identify property value occurred in the sentence and then property name association will be carried out. The property name association is used to identify its property value.

3.1.6 Action relation extractor

Action extractor phase will identify all actions according to the rules made. Once the action has been extracted the next step is find the related concepts as subject and object. Subject and object found for extracted relation is divided into two categories. Simple sentence: simple sentences are those which contain pattern like subject–verb–object i.e. [NP1 VP1 NP2] and compound sentence.

3.1.7 Graph representation of ontology

This step provides the visual representation of generated data that is stored in the relational databases. Concepts of domain will form the nodes. Relationship between concepts will be shown with help of action. Edges in graph will be labelled by actions and properties names. Property values also form nodes of graph.

3.1.8 OWL representation

The generated ontology can be represented in OWL. For this ontology generated in previous steps will be created by an ontology building editor. With the help of used tool OWL representation will be produced.

4 Corpus based analysis

In order to construct the syntactic–semantic relationships. We take a piece of text from computer domain and then it is given as input to the Stanford parser. For example, “Computer is general purpose device that can be programmed to carry out arithmetic or logical operations. Since sequence of operations can be changed, computer can solve more than one kind of problem. Computer consists of at least one processing element, central processing unit and memory. Processing element carries out arithmetic and logic operations, and sequencing and control unit can change order of operations” is one instance of the text to be manually analysed.

Corresponding to this text the parser generates following structure which is a tagged text using Perm Treebank tagging scheme.

computerlNN is/VBZ general/JJ purpose/NN device/N’N thatlWDT can/MD beIVB programmedlVBN tolTO carry/Vll out/RP arithmetic/NN or/CC 10gicINN operations/NbiS since/IN sequence/NN oflIN operationslNNS canlMD beIVB changedIVBN,I, computer/N’N canlMD solveIVB more/]]R thanIIN one/CD kindlNN of/IN problem/NN.,(computer/N’N consistslVBZ oflIN at/IN least/]]S one/CD processing/Nbl element/N’N,I, central/Ll processing/VBG unitlNN and/CC formlNN oflIN memorylNN.f. ProcessinglVBG element/NN carriesIVBZ,out/RP, arithmeticINN, and/CC logicINN operations/N’N S,I, and/CC sequencingIVBG and/CC control/INN unitINN thatIWDT can/MD changeIVB orderlNN of/IN operations/N’N S.

To restore the singular form, tagged text after parsing undergoes the process of morphological analysis using the following rule.

Rule M1 If current token tag is equal to ‘‘NNS’’ then get the singular form of current token.

4.1 Sentence level analysis

The tagged text after morphological analysis is now used to perform sentence level analysis to extract the domain specific concepts and the relationship among them. Each sentence undergoes the following phases:

- Pattern/rule identification for concepts, properties and actions
- Building rules for nested concept generation
- Building rules for hierarchical relationship
- Building rules for properties
- Building rules for actions which actually form relations between concepts.

The following sub-sections gives the detailed view about these phases.

4.1.1 Pattern identification for concepts, properties and actions

In this phase, the tagged form of each sentence is taken and from the tag type the concept, properties and actions are identified. For example, corresponding to following sentence, the identified concepts, properties and action are shown in Table 1 in Appendix.

Sentence 1 Computer/NN is/VBZ general/JJ purpose/N'N device//NN that/WDT can/MD be/VB programmed/VBN to/TO carry/VB out/RP arithmetic/NN or/CC logic/NN operation/NN

For the given a sentence, the identified patterns for concepts and actions are listed along with properties and assign property names. The pattern is enclosed with "[]". In this way analysis is done for all sentences in the text and patterns/rules for action relation have been recognized, On the basis of analysis, the integrated list of patterns/rules identified for noun phrase used to form concepts is given in Table 2 in Appendix and pattern/rules for action relation is given in Table 3 in Appendix.

4.1.2 Rules for nested concept generation

It may be possible that more than one concept can be generated by a single noun phrase. Many valid concepts may be generated by a single valid concept. The rules for generating all concepts existed in a single noun phrase are given in Table 4 in Appendix. Make concept is a semantic action which is taken when the given pattern is found. Word(X) specifies the word in the noun phrase corresponding to tag X.

4.1.3 Rules for hierarchical relationship

Hierarchical relationship is a type of relationship which is often called "Is A" relationship. In the analysis, two types of hierarchical relationship have been found.

1. Inherent relationship in nested concepts.
2. Hierarchical relationship defined by linking verb "is".

Inherent relationship-formation of nested concepts shows the inherent relationship between concepts. On the basis of analysis it is found that any noun in the noun phrase is related to the noun to its right in the same noun phrase. The inherent relationship among concepts generated from noun phrase "Electronic Digital Computer" is shown in Fig. 3.

There are two inherent relationships shown in Fig. 3. First digital computer is sub-class of Computer; second Electronic Digital Computer is sub-class of Digital Computer. Such type of hierarchical relationship will be extracted with help of following rule.

Rule H1 If noun phrase pattern [NN I NN2] is found and token corresponding to [NN2] forms a valid concept C I and tokens corresponding to [NN I NN2] together form a valid concept C2 then set concept Cl as super-class of concept C2.

Rule H2 If noun phrase pattern [JJ NN] is found and token corresponding to [NN] forms a valid concept Cl and tokens corresponding to [JJ NN] together form a valid concept C2 then set concept Cl as super-class of concept C2.

In this way the rules have been defined for a noun phrase sequences to capture the inherent relationship

Hierarchical relationship: in most of sentences it is observed that two noun phrases connected with linking verb "is" shows the hierarchical relationship. E.g. Cow is animal. In the given example it is easily derived that cow is a type of animal Cow belongs to animal class. By identifying the pattern like [NP 1 is NP2] in the sentence, it is easy to capture existed hierarchical relationship. Concept generated from.NPI is always has "sub-class of' relation with concept generated from NP2.

4.1.4 Rules for properties

Properties describe the general and specific characteristics of individuals. Properties identified here are mainly

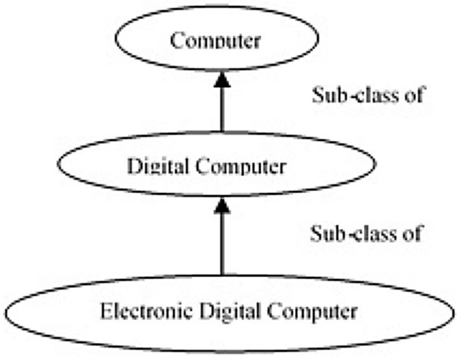


Fig. 3 Inherent relationship in nested concept

datatype property. Based on the position of the word forming property value, the property extraction can be of two types:

1. When the word forming property value is present anywhere in the sentence except concept.
2. When the word forming property value is the part of concept itself. E.g. small chair

E.g. Chair is small. In a given sentence the information derived by human is “chair is small in size”. Small refers to size here assuming “small chair” represents the class or concept of all the small sized chairs. Here word small also indicates the size of chairs. On the basis of manual analysis it is concluded that the adjectives are used to describe the concepts. When property value occurs in the sentence, the proper property name is assigned. On the basis of analysis the following rule is made to identify such property value.

Rule P1 If (any adjective is found) then get its property name.

After finding the property the next step is to find the concept to which this property describes. It is easy to find the concept related to the extracted property, when the property is the part of concept itself so rule for this is:

Rule P2 If the property value is the part of noun phrase and that noun phrase generates a valid concept, then the concept generated from that noun phrase would be the candidate concept.

In first case, when the property value can be anywhere in the sentence, the main concept of the sentence or sub-sentence will be the candidate concept to which property belongs. Although most of the adjectives form property value, it is found in the analysis some noun also forms the property value. Some nouns also will be tested for the same so a new rule is derived for this

Rule P3 If (a noun matches to the property value) then assign the property name to it.

4.1.5 Rules for actions

Each sentence is analysed to find actions. Actions basically are relation in an ontology which shows the relationship between two concepts. In ontology these actions is used to build Object Type property. On the basis of analysis, it is found that an action is performed by someone and also performed on something or for something. The doer of action is known as subject and the something on which the action is being performed is known as object. The following section shows the process of finding subject and object of action. On the basis of manual analysis it is found that sentences can be categorized into two categories:

1. Simple sentence
2. Compound sentence

Simple sentence is of the form subject-verb-object. The following example of sentence describes this form of sentence as well as the process of finding action, subject and object of action Identification of subject and object for an identified action in a simple sentence with help of a sample sentence is shown in Fig. 4.

Sentence Mechanical/JJ analog/JJl computer/NN were/VBD used/VBN for/IN military/JJ application/NN./.

So rules for subject object identification are:

Rule A1 Left hand side of the action contains subject.

Rule A2 Right hand side of the action contains object.

Compound sentence: compound sentence is a sentence which consists of two or more simple sentences connected by conjunction elements, for example, that, but, since, while, etc. e.g.

Computer is a general purpose device **that**
S1
can be programmed to carry out arithmetic or logical operations.
S2

There are two actions found in above sub-sentence: can be and to carry out. After manual analysis, it is decided that any pattern of the fonn [MD VB VBN] will be processed with the help of following rule:

Rule A3 If pattern [MD VB VBN] found then tokens corresponding to pattern [MD VB] will form property together and token corresponding to tag VBN will from property value under the property formed by the same partial pattern

After manual analysis, it is decided that any pattern of the form [To VB RP] will- be processed with the help of following rule:

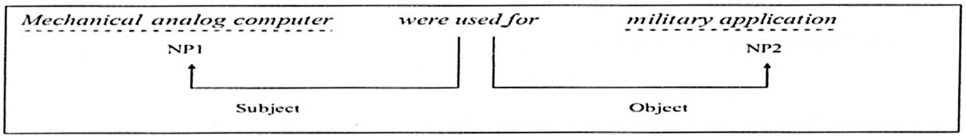


Fig. 4 Showing subject–object for identified action in a simple sentence

Rule A4 If pattern [TO VB RP] found Then action will be formed as "can" along with the tokens corresponding to partial pattern [VB RP] together.

After getting the action present in the sentence the concept which is doing that action, will be identified. In this case the following rule will help to find the doer of the action.

Rule A5 For an action present in the sentence, if there is no subject given then the subject of previous sentence will be the subject of that action.

4.2 Dictionary

Dictionary is used as a validation mechanism to validate concept and relationship amongst them which is the main goal of ontology development. So after the extraction of concepts and properties of concepts, the knowledge must be stored or incorporated in the form of dictionary. These valid concepts residing in dictionary can be used to get more relevant and close to (logical–physical) real world concepts.

4.2.1 Dictionary for valid concepts

In the proposed method, all the valid concepts identified are kept in a dictionary in form of relational database. This dictionary is called VCD that will be used for validating the extracted concepts by the proposed system The dictionary containing all the valid concepts identified is shown in Table 5 in Appendix.

4.2.2 Dictionary for valid properties

In the proposed method, all the valid properties values identified are kept in a dictionary in form of relational database. The dictionary contains all identified valid property values withtheir assigned names. This dictionary shown in Table 6 in Appendix is called VPD that will be used for assigning proper name to property value when it occurs in text.

5 Implementation of the proposed system

The proposed system has been implemented using JAVA platform and DBMS MS-Access. The Java API for Stanford parser is integrated with the rest of the modules written in Java. Graph language DOT (Appendix) is used to write graphs and representation of graphs. For displaying ontology in form of graph, an editor GVEdit for Graph Viztool is used. Finally Ontology representation in OWL format is done using SWOOP editor.

5.1 Results

In this section the result produced by the system for a given input is shown Input text given to the system.

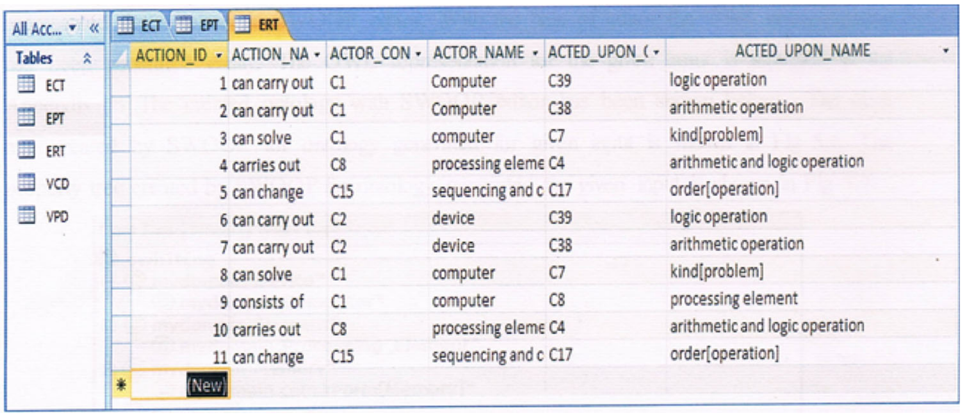
> Computer is general purpose device that can be programmed to carry out arithmetic or logic operations. Since sequence of operations can be changed, computer can solve more than one kind of problem. Computer consists of at least one processing element, central processing unit and form of memory. Processing element carries out arithmetic and logic operations, and sequencing and control unit that can change order of operations.

5.1.1 Ontology generated by the system for the given input

System processes the given input text and produces various intermediate results for all the phase discussed above. The final ontology stored in the databases ECT, EPT, ERT is shown below. Database ECT shown in Fig. 5 gives the details of the concepts and hierarchical relationship obtained.

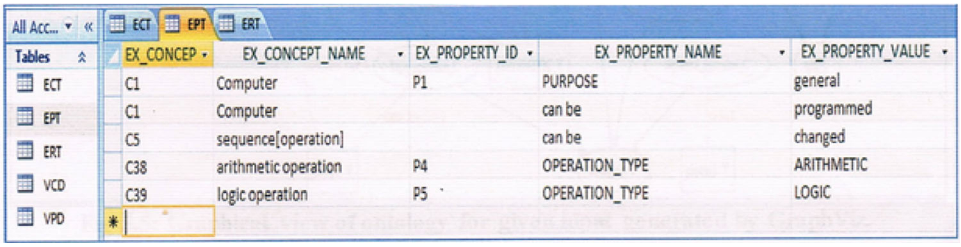
The database EPT shows the properties extracted by the system the snapshot of database EPT for the given input is shown in Fig. 6.

The following database ERT shows the relations extracted by the system The snapshot of the database ERT is shown in Fig. 7.



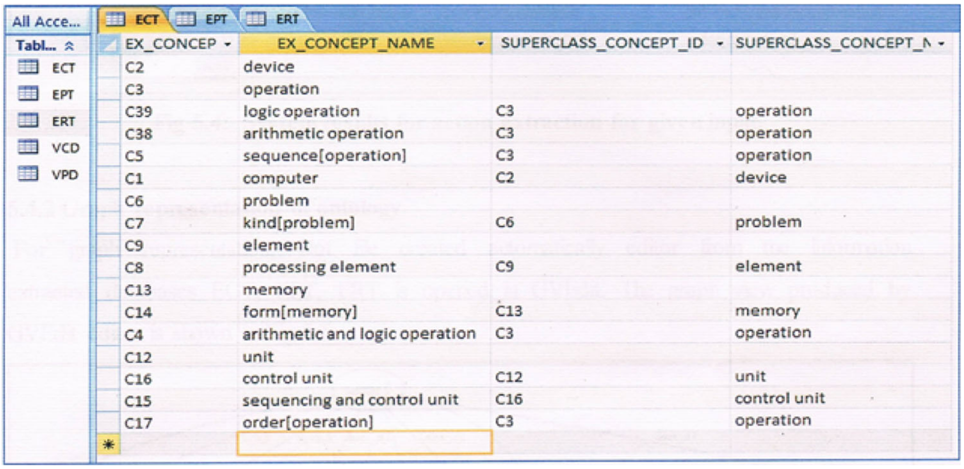
ACTION_ID	ACTION_NA	ACTOR_CON	ACTOR_NAME	ACTED_UPON_(	ACTED_UPON_NAME
1	can carry out	C1	Computer	C39	logic operation
2	can carry out	C1	Computer	C38	arithmetic operation
3	can solve	C1	computer	C7	kind[problem]
4	carries out	C8	processing eleme	C4	arithmetic and logic operation
5	can change	C15	sequencing and c	C17	order[operation]
6	can carry out	C2	device	C39	logic operation
7	can carry out	C2	device	C38	arithmetic operation
8	can solve	C1	computer	C7	kind[problem]
9	consists of	C1	computer	C8	processing element
10	carries out	C8	processing eleme	C4	arithmetic and logic operation
11	can change	C15	sequencing and c	C17	order[operation]

Fig. 5 System results for concept extraction for given input



EX_CONCEP	EX_CONCEPT_NAME	EX_PROPERTY_ID	EX_PROPERTY_NAME	EX_PROPERTY_VALUE
C1	Computer	P1	PURPOSE	general
C1	Computer		can be	programmed
C5	sequence[operation]		can be	changed
C38	arithmetic operation	P4	OPERATION_TYPE	ARITHMETIC
C39	logic operation	P5	OPERATION_TYPE	LOGIC

Fig. 6 System results for property extraction for given input



EX_CONCEP	EX_CONCEPT_NAME	SUPERCLASS_CONCEPT_ID	SUPERCLASS_CONCEPT_N
C2	device		
C3	operation		
C39	logic operation	C3	operation
C38	arithmetic operation	C3	operation
C5	sequence[operation]	C3	operation
C1	computer	C2	device
C6	problem		
C7	kind[problem]	C6	problem
C9	element		
C8	processing element	C9	element
C13	memory		
C14	form[memory]	C13	memory
C4	arithmetic and logic operation	C3	operation
C12	unit		
C16	control unit	C12	unit
C15	sequencing and control unit	C16	control unit
C17	order[operation]	C3	operation

Fig. 7 System result for action extraction for given input

5.1.2 Graph representation of ontology

For graph representation, Dot file created automatically editor from the informationextracted databases ECT, EPT, ERT is opened in GVEdit. The graph view produced by GVEdit editor is shown in Fig. 8.

5.1.3 OWL representation using Swoop tool

The generated ontology in form of graph can be exported on any ontology editor. We have represented the ontology in SWOOP editor. With the help of editor the OWL representation has been generated easily. The created ontology with SWOOP editor (http://www.mindswap.org/2004/SWOOP) has been shown below. The class tree created by SWOOP for ontology generated for given input is shown in Fig. 8. The property tree created by SWOOP for ontology generated for given input is shown in Fig. 9.

The property tree created by SWOOP for ontology generated for given input is given below (Fig. 10).

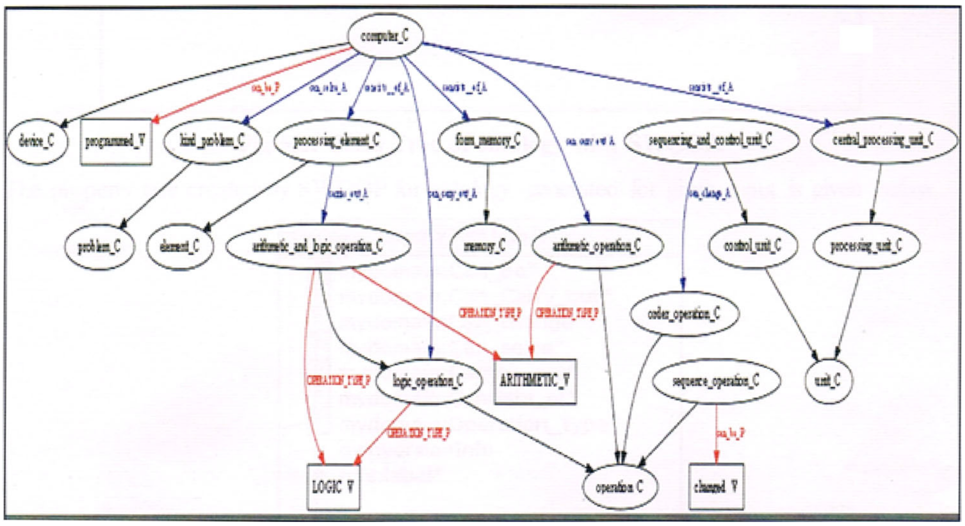
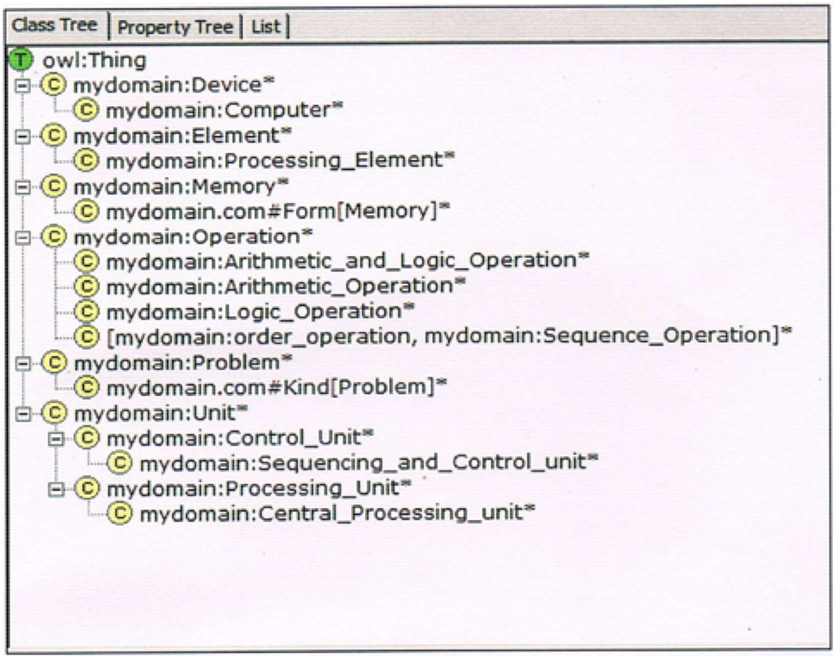
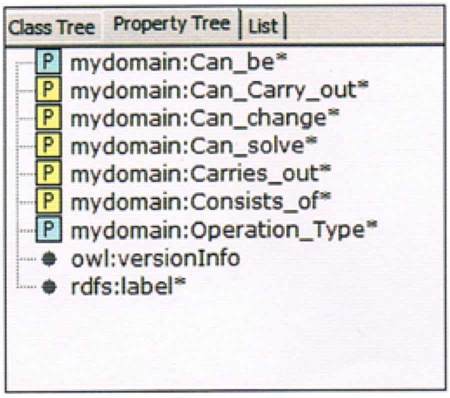


Fig. 8 Graphical view of ontology for given input generated by Graph Viz



```
T owl:Thing
⊟ C mydomain:Device*
    C mydomain:Computer*
⊟ C mydomain:Element*
    C mydomain:Processing_Element*
⊟ C mydomain:Memory*
    C mydomain.com#Form[Memory]*
⊟ C mydomain:Operation*
    C mydomain:Arithmetic_and_Logic_Operation*
    C mydomain:Arithmetic_Operation*
    C mydomain:Logic_Operation*
    C [mydomain:order_operation, mydomain:Sequence_Operation]*
⊟ C mydomain:Problem*
    C mydomain.com#Kind[Problem]*
⊟ C mydomain:Unit*
  ⊟ C mydomain:Control_Unit*
      C mydomain:Sequencing_and_Control_unit*
  ⊟ C mydomain:Processing_Unit*
      C mydomain:Central_Processing_unit*
```

Fig. 9 Class tree of ontology with SWOOP



```
P mydomain:Can_be*
P mydomain:Can_Carry_out*
P mydomain:Can_change*
P mydomain:Can_solve*
P mydomain:Carries_out*
P mydomain:Consists_of*
P mydomain:Operation_Type*
● owl:versionInfo
● rdfs:label*
```

Fig. 10 Property tree of ontology with SWOOP

6 Conclusion and future work

In this paper, we have proposed and implemented a system to find semantics of English text in the form of an ontology. The proposed approach is novel as it uses statistical methods, though indirectly through Stanford parser, and computational linguistic techniques. In order to process a given text, various rules and dictionaries are constructed from the carpus. The performance of the proposed system is evaluated on small set of text document and the system is able to extract the semantic information corresponding to the text in the form of ontologies.

However, the performance and the robustness of the system depends on the richness of rules and size dictionary of concepts. At presents, the rules and dictionary is constructed for a limited set of text. Therefore, there is huge scope of enriching the set of rules and concepts from a larger and wide set of texts. Also the present work, is based on the manual analysis. But in future, the base rules and dictionary may built using automated approach.

Compliance with ethical standards

Conflict of interest None.

Appendix

See Tables 1, 2, 3, 4, 5 and 6.

Table 1 Identified patterns for concepts, properties and action for Sentence 1

Concept name	Pattern
Computer	[NN]
Device	[NN]
Operation	[NN]
Arithmetic operation	[JJ NN]
Logic operation	[JJ NN]
Property value	**Property name**
General	Purpose
Programmed	Can be
Action name	**Pattern**
Carry out	[TO VB RP]

Table 2 Integrated list of patterns for noun phrases

Noun phrase pattern	Concept example
[NN]	Computer, Memory
[NN IN NN]	Sequence of operations, Kind of Problem
[NN NN]	Processing Element
[JJ NN NN]	Central Processing Unit
[NN CC NN NN]	Sequencing control unit

Table 3 Integrated list of patterns for actions

Action pattern	Action example
[MD VB VBN]VB	Can be programmed, can be changed
[MD VB VBN IN]	Can be used for
[MD VB VBN RP]	Can be carried out
[TO VB RP]	To carry out
[TO VB IN]	To get in, To use for
[MD VB]	Can solve, Can change
[MD VB IN]	Can consist of
[MD VB RP]	Can carry out
[VBZ RP]	Carries out
[VBP RP]	Carry out
[VBZ IN]	Uses for, Consist of
[VBP IN]	Fit into
[VBD VBN IN]	Where developed between, where used for
[IN]	Any preposition anywhere in the sentence(except "of") related to the main verb of sentence
[VBZ]	Occupies
[VBG]	Consuming
[VBZ VBG RP]	Is carrying out
[VBP VBG RP]	Are carrying out
[VBZ VBG IN]	Is consisting of
[VBP VBG IN]	Are consisting of

Table 4 Rules for generating nested concepts

Rule N1: if P = [(JJ)*NN] found and length	make_concept(word(NN));
	make_concept(word/P_{i-1}) + make_concept (P_i)
	where i ← L to 1

Rule N2: if P = [(NN) (NN)] found and length of pattern is L then	
Rule N3: if P = [(JJ) (NN) (NN)] found and length of pattern is L then	

Rule N4: if P = [NNI CC NN2 NN3] and Word(CC) = "OR"	Make_concept(word(NN 3))
	Make_concept(word(NN2) + Word(NN3))
	Make_concept(word(NN 1) + word(NN3))

Rule N5: if P = [NNI CC NN2 NN3] and Word(CC) = "AND"	Make_concept(word(NN3))
	Make_concept(word(NN 1) + word(CC) + word(NN3))

Rule N6: if P = [NNI IN NN2] and Word(lN) = "OF"	Make_concept(word(NN 1) + "["+word(NN2)+ "]")

Table 5 Dictionary of valid concepts for computer domain

CONCEPT_ID	Valid concept name
1	COMPUTER
2	DEVICE
3	OPERATION
4	ARITHMETIC OPERATION
5	LOGIC OPERATIONS
6	SEQUENCE [OPERATION]
7	PROBLEM
8	KIND [PROBLEM]
9	PROCESSING ELEMENT
10	ELEMENT
11	CENTRAL PROCESSING ELEMENT
12	MEMORY
13	FORM [MEMORY]
14	SEQUENCING AND CONTROL UNIT
15	CONTROL UNIT
16	SEQUENCING UNIT
17	ORDER {OPERATION]
18	FIRST ELECTRONIC DIGITAL COMPUTER
19	ELECTRONIC DIGITAL COMPUTER
20	DIGITAL COMPUTER
21	ELECTRONIC COMPUTER
22	MECHANICAL ANALOG COMPUTER
23	ANALOG COMPUTER
24	APPLICATION
25	MILITARY APPLICATION
26	MOBILE DEVICE
27	MOBILE COMPUTER
28	BATTERY
29	SMALL BATTERY

Table 6 Dictionary of valid properties for computer domain

Property value	Property name
General	PURPOSE
Arithmetic	OPERATION TYPE
Logic	OPERATION TYPE
Large	SIZE
Small	SIZE
Military	APPLICATION TYPE
Digital	SIGNAL TYPE
Analog	SIGNAL TYPE
Mechanical	TECHNOLOGY_TYPE 1
Electronic	TECHNOLOGY_TYPE 1
Integrated circuit	TECHNOLOGY_TYPE 2
Personal	USAGE_TYPE
Modem	ERA_TYPE
Early	ERA_TYPE
External	SOURCE_TYPE
Peripheral	DEVICE_TYPE
Mobile	MOBILITY_TYPE

References

ADen L (2004) Natural language understanding. Pearson Publication, New York

Agirre E, Ansa O, Hovy E, Martinez D (2000) Enriching very large ontologies using the WWW. In: Proceedings of the ontology learning workshop, ECAl, Berlin, Germany

Allen JF (1993) Natural language, knowledge representation and logical form. In: Bates M, Weischedel R (eds) Challenges in natural language processing. Cambridge University Press, Cambridge

Bedini I, Nguyen B (2007) Automatic ontology generation: state of the art. In: PRiSM Laboratory Technical Report. University of Versailles

Cho M, Kim H, Kirn P (2006) A new method for ontology merging based on concept using wordnet. In: Advanced communication technology, 2006. ICACT 2006. The 8th international conference, vol 3, p 157311576

Chomsky N (2002) Theory of syntactic structures. ISBN-13: 978-1614278047

Dorr BJ, Jordan PW, Benoit HW (1998) A survey of current paradigms in machine translation. Technical Report: LL\MP-TR-027, UMIACS-TR-98-72, CS-TR-3961, University of Maryland, College Park, December 1998

Fromkin V et al (2000) Linguistics: an introduction to linguistic theory, Chapter 3.2 (Constituent Order, Case Marking and Thematic Roles). Blackwell Publishing, Hoboken

Gruber TR (1995) Toward principles for the design of ontologies used for knowledge sharing. Int J Hum Comput Stud 43:907–928

http://ww.w3.org/2003/08/owlfaq

Jurafsky D, Martin J (2000) Speech and language processing: an introduction to natural language processing, computational linguistics, and speech recognition upper saddle river. Prentice Hall, New Jersey

Kietz J, Maedche A, Volz R (2000) A method for semi-automatic ontology acquisition from a corporate intranet. In: Proceedings of EKA W-2000 workshop ontologies and text, Juan-Les-Pins, France

Kong H, Hwang M, Kim P (2006) Design of the automatic ontology building system about the specific domain knowledge. In: Advanced communication technology, 2006. ICACT 2006. The 8th international conference, vol 2

MiUer GA (1995) WORDNET: a lexical database for English. Commun ACM 1:39–41

Moldovan DI, Girju R (2000) Domain-specific knowledge acquisition and classification using wordnet. In: Proceedings of the thirteenth international florida artificial intelligence research society conference, pp 224/228. AAAI Press, p 81

Noy NF, McGuinness DL. Ontology Development 101: a Guide to creating your first Ontology. http://protege.stanford.edu/publications/ontology_development/ontology101.pdf